

УДК 658.54 + 621.3.049.771.14

## **Structure Development of Information Design Support System for «System on a Chip» Circuits Based on Subject-Oriented Ontologies**

**Oleg V. Drozd\* and Denis V. Kapulin**

*Siberian Federal University  
79 Svobodny, Krasnoyarsk, 660041, Russia*

Received 18.06.2016, received in revised form 23.08.2016, accepted 19.11.2016

---

*This article discusses the design stages and transferring into manufacturing of microelectronic products based on integrated circuits «System on a chip». The proposed structure of the Product Data Management (PDM) system, which is a key element of Integrated Information Environment that meets the requirements of microelectronic manufacturing. Analysis and design of the PDM-system's proposed structure are made using the provisions of the subject-oriented ontology. Formed requirements for the composition and content of the documents included in the information support of automated process control systems microelectronic production lifecycle, which has a hierarchical structure. Results of the study can be used in the development of the architecture circuits «System on a chip» product data management electronic system to integrate all stages of the life cycle and the organization of Integrated Information Environment of the design or design and production organizations.*

*Keywords: PDM, information support, system on a chip, microelectronics, integrated information environment, subject-oriented ontology.*

---

Citation: Drozd O.V., Kapulin D.V. Structure development of information design support system for «System on a chip» circuits based on subject-oriented ontologies, J. Sib. Fed. Univ. Eng. technol., 2016, 9(7), 1079-1096. DOI: 10.17516/1999-494X-2016-9-7-1079-1096.

---

© Siberian Federal University. All rights reserved

\* Corresponding author E-mail address: olvldrozd@gmail.com

## **Разработка структуры информационной системы поддержки проектирования сверхбольших интегральных схем класса «Система на кристалле» с использованием предметно-ориентированных онтологий**

**О.В. Дрозд, Д.В. Капулин**

*Сибирский федеральный университет  
Россия, 660041, Красноярск, пр. Свободный, 79*

---

*В статье рассматриваются этапы проектирования и передачи в производство изделий микроэлектроники на сверхбольших интегральных схемах класса «Система на кристалле». Предложена структура системы управления данными об изделиях (PDM-система), являющаяся ключевым элементом единого информационного пространства, отвечающего требованиям микроэлектронного производства. Анализ и проектирование предлагаемой структуры PDM-системы выполнены с использованием положений предметно-ориентированной онтологии. Сформированы требования к составу и содержанию документов, входящих в информационное обеспечение автоматизированной системы управления процессами жизненного цикла микроэлектронного производства, имеющей иерархическую структуру. Результаты исследования могут быть использованы при разработке архитектуры системы управления данными электронных изделий класса «Система на кристалле» для интеграции всех этапов жизненного цикла и организации единого информационного пространства проектной или проектно-производственной организации.*

*Ключевые слова: система управления данными, информационное обеспечение, система на кристалле, микроэлектроника, единое информационное пространство, предметно-ориентированная онтология.*

---

### **Введение**

Процессы проектирования и производства изделий микроэлектроники на современном этапе своего развития нуждаются в децентрализации и унификации, что связано с непрерывным развитием информационных технологий и повсеместной интенсивной информатизацией проектной деятельности. Для унификации проектных решений необходима декомпозиция процесса проектирования сверхбольших интегральных схем (СБИС) с применением заранее разработанных и многократно используемых схмотехнических блоков. Такой подход к проектированию реализуется в методологии «Система на кристалле» (СнК, *System on a Chip, SOC*), а составные части разрабатываемой в рамках такой методологии микросистемы называются сложнофункциональными (СФ), или *IP*-блоками (*Intellectual Property* – интеллектуальная собственность) [1].

Для успешной деятельности комплексной проектно-производственной среды задачи проектирования и разработки СнК должны быть тесно увязаны со средствами управления такими процессами. Объекты знаний должны распространяться автоматически или по команде между программами, сотрудниками и коллективами подразделений. Главная задача формирования такой цепи распространения – интеграция знаний различных источников для обеспечения ин-

формационного взаимодействия отдельных работников и коллективов в рамках единого информационного пространства (ЕИП) [2].

Эффективная организация единого информационного пространства способствует снижению временных и финансовых издержек процессов проектирования и производства, но при этом требует значительных усилий для своего развертывания и внедрения, что обусловлено сменой устоявшихся механизмов управления. Информационное взаимодействие внутри организации должно быть поддержано автоматизированными информационными системами сбора, обработки и анализа разнородных данных [3]. Кроме того, организация ЕИП предполагает работу участников процесса проектирования СБИС СнК в единой понятийной среде по единым правилам адресации субъектов и объектов ЕИП, а также согласование содержания информационного и информационно-лингвистического обеспечения объектов ЕИП и поддержание их в актуальном состоянии.

### Постановка задачи

Эффективность проектирования и производства СБИС СнК на сегодняшнем этапе развития подходов к управлению качеством процессов управления возможно обеспечить за счет интеграции всех этапов проектирования и подготовки к производству, а также посредством их объединения с обеспечивающими и вспомогательными работами в едином информационном пространстве. Известен целый ряд примеров внедрения средств ЕИП на промышленных предприятиях [4–6], однако большинство из них ориентировано на металлоемкие области машиностроения, что делает непригодными существующие методики для их прямого внедрения на предприятиях – производителях электронных устройств класса «Система на кристалле». При этом основной проблемой является противоречивость логической структуры машиностроительных систем по отношению к информационному обеспечению автоматизированных систем, ориентированных на микроэлектронное производство.

Микроэлектронные дизайн-центры и фабрики, реализующие единую согласованную информационную инфраструктуру, относятся к категории «разумных» производств. «Разумное» производство (*smart foundry*) включает в себя не только технологический участок, но и центр проектирования и прототипирования, службу поддержки и согласования требований заказчиков. При этом допускается использование уникальных согласованных модификаций технологических процессов и маршрутов проектирования для каждого проекта. Такое производство существенно дороже, но обеспечивает рентабельный выпуск уникальных изделий с высокой добавленной стоимостью малой серией [7].

Для эффективной организации ЕИП «разумного» производства изначально следует разработать структуру системы управления данными об изделии, являющейся ключевым элементом ЕИП. Важный предварительный этап перед разработкой – исследование информационных процессов предметной области и формирование иерархической структуры системы управления данными об изделии (СУД, *Product Data Management – PDM*). В качестве эффективного средства проведения таких исследований, связанных с проектированием иерархических структур данных, часто используется аппарат предметно-ориентированных онтологий [8–10].

Применение аппарата предметно-ориентированных онтологий позволит сформировать иерархическую структуру элементов предметной области, однозначно определить их свойства и взаимоотношения. С использованием такого аппарата совместно с другими средствами проектирования будет возможно получить непротиворечивую структуру *PDM*-системы, обладающую полнотой и при этом не отличающуюся избыточностью.

### **Особенности проектирования СБИС класса «Система на кристалле»**

Прежде чем перейти непосредственно к рассмотрению единой модели представления данных об изделии и структуры *PDM*-системы, необходимо кратко изложить маршрут проектирования СБИС класса «Система на кристалле». Весь цикл проектирования можно представить в виде совокупности четырех основных этапов [11, 12]:

1. Системное проектирование – определение основных эксплуатационно-технических свойств будущей системы: требуемое быстродействие; допустимая потребляемая мощность; время, необходимое для разработки, и т. д. На основании этих свойств создается системная спецификация, которая может выступать частью технического задания на разработку системы.
2. Функциональное проектирование – разработка и верификация синтезируемой *RTL*-модели (*Register Transfer Level*) на уровне регистровых передач.
3. Логическое проектирование – разработка логической схемы СнК.
4. Топологическое проектирование – разработка и верификация топологии СБИС.
5. Этап верификации и подготовки к производству.

На этапе системного проектирования разрабатывается поведенческая модель СнК и определяется состав СФ-блоков. Исполняемые спецификации представляются в определенном формате на языках C, C++, *Verilog* и *VHDL*.

Разработка СнК начинается с анализа задач и требований, а также написания системной спецификации. При этом определяются основные эксплуатационно-технические свойства СнК, такие как требуемое быстродействие и допустимая потребляемая мощность.

Алгоритм работы СнК на уровне математического описания создается на основе разработанной системной спецификации. Производится математическое моделирование разработанных алгоритмов функционирования СнК, оценка требуемого быстродействия, разрядности и оценка формата представления внутренних данных. Также при необходимости может производиться синтез наборов данных (сигналов), предназначенных для тестирования схемотехнических решений СнК.

На основе алгоритма работы СнК создается поведенческая модель системы на уровне СФ-блоков в виде блок-схемы, отражающей принцип взаимодействия СФ-блоков в составе СнК и включающей их основные параметры. Для верификации поведенческой модели создают проект тестового окружения (*Testbench*). Оно включает в себя тестовые последовательности, генераторы входных сигналов и средства отображения выходной информации. На основе тестового окружения впоследствии разрабатываются тестовые последовательности для верификации проекта на нижних уровнях проектирования, а также для функционального тестирования опытных образцов. После этого проводят верификацию поведенческой модели путем компьютерного моделирования с помощью специальных программных средств.

Далее принимается решение о том, какие СФ-блоки поведенческой модели будут впоследствии реализованы на аппаратном уровне, а какие – на программном на основе встроенных в СнК процессорных ядер; разрабатывается интерфейс между аппаратным и программным обеспечением и определяется общая аппаратная архитектура СнК. Разрабатывается спецификация и определяются приобретаемые, имеющиеся в наличии и вновь разрабатываемые СФ-блоки. В итоге создается набор спецификаций на разработку как программного обеспечения, так и каждого аппаратно реализуемого СФ-блока.

При совместной программно-аппаратной верификации проверяется функционирование аппаратного обеспечения, разрабатываемого в соответствии с имеющимися спецификациями, под управлением встроенного программного обеспечения в реальном масштабе времени. В качестве аппаратного обеспечения используются исполняемые спецификации аппаратно реализуемых блоков, в качестве программного – прототип прикладного программного обеспечения (ПО), развертываемого на базе СнК (например, прикладное ПО для цифровой обработки сигналов на базе операционных систем реального времени *Unison* и других *Unix*-подобных систем для архитектур *MicroBlaze* и *Cortex*).

При выборе архитектуры одним из ключевых параметров является низкое энергопотребление микросхемы. Исходя из этого, производят подбор СФ-блоков и разделение на аппаратный и программный уровни. Поэтому необходимо получить и систематизировать полную техническую информацию о покупных и имеющихся в наличии СФ-блоках, так как их характеристики могут существенно повлиять на архитектуру микросхемы.

### **Концептуальная модель единого информационного пространства**

Рассмотренные этапы маршрута проектирования СБИС класса СнК обладают высокой степенью связности и информационной взаимозависимости. По этой причине для организации и эффективного управления процессом промышленного проектирования СнК необходимо развитие и внедрение средств и систем управления проектными данными в рамках единого информационного пространства. Как показано в [13, 14], при построении ЕИП повсеместно применяется главный принцип, согласно которому информация, однажды возникшая на каком-либо этапе проектного цикла, сохраняется в ЕИП и становится доступной всем участникам этого и других этапов работ с учетом прав доступа к информации. Это позволяет избежать дублирования, несанкционированного изменения данных, ошибок и сократить временные и трудовые затраты.

Основой ЕИП, включающей в себя механизмы сбора, обработки, структуризации и анализа данных, является *PDM*-система. Такая система управления данными позволяет организовать управление проектированием изделия, его производством и другими процессами с точки зрения их информационной поддержки за счет консолидации информации всех используемых на предприятии прикладных автоматизированных систем и преобразования разрозненных данных в информационное обеспечение ЕИП. Результатом такого подхода служит эффективное управление информацией за счет повышения доступности данных об изделии, интегрированных в единую информационную модель.

Концептуальная модель ЕИП (рис. 1) включает в себя следующие составляющие:

- участники проектного цикла – поставщики информации;
- *PDM*-система, реализующая унифицированный обменный формат единой модели данных и предоставляющая платформу для информационного взаимодействия между поставщиками и потребителями информации;
- участники проектного цикла – потребители информации.

Основными источниками проектных данных в едином информационном пространстве микроэлектронного производства выступают *CAx*-системы (*CAD/CAE/CAM/CAPP*), системы имитационного моделирования, а также интегрированные среды разработки *HDL*-кода описания аппаратной части и программного кода прикладного ПО СнК. Системы *CAx* в микроэлектронике в основном представлены продуктами компаний *Mentor Graphics* и *Cadence Design Systems*. Интегрированные среды разработки поставляются как вышеупомянутыми компаниями, так и компаниями-разработчиками ПЛИС, такими как *Altera (Quartus)*, *Xilinx (Vivado Design Suite)* и *Lattice (Lattice Diamond)*. Наиболее популярная система имитационного моделирования – пакет *MATLAB* (с использованием пакетов расширения *Communications System Toolbox*, *HDL Coder*, *HDL Verifier*).

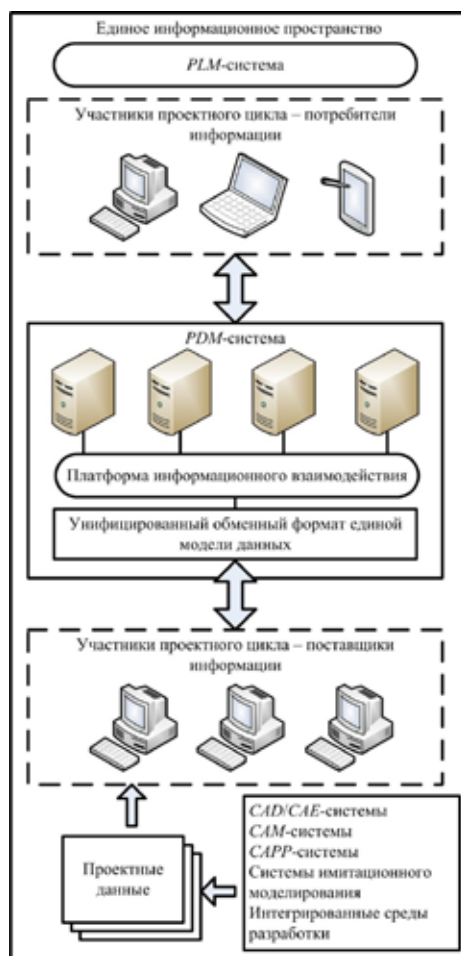


Рис. 1. Концептуальная модель ЕИП

Участники проектного цикла – поставщики информации – передают полученные на текущем этапе проектного цикла (ПЦ) данные участникам-потребителям, которые могут быть расположены как на текущем этапе ПЦ, так и на последующем или предыдущем этапе ПЦ. Причем поставщик информации может также выступать и в роли потребителя информации от других участников проектного цикла. Приведение разнородных данных от программных систем сторонних разработчиков к унифицированному единому формату, хранение данных и информационное взаимодействие между поставщиками и потребителями осуществляются *PDM*-системой.

При проектировании информационного пространства необходимо выработать формальное описание единой модели данных, а также описание способов преобразования циркулирующей в системе информации из форматов, используемых в программном обеспечении сторонних разработчиков, в унифицированный формат единой модели данных. Кроме того, необходимо сформировать структуру единой модели данных. Разработка единой модели данных *PDM*-системы является ключевым этапом проектирования ЕИП, так как позволяет формализовать и сопоставить в виде единой структуры атрибуты файлов проектных данных, что необходимо для формирования структуры и требований к унифицированному обменному формату данных, а также функционалу программных средств обеспечения взаимодействия между *PDM*-системой, пользователями и сторонним ПО. Использование аппарата предметно-ориентированных онтологий позволяет выполнить указанные требования, убрать противоречивость и избыточность структуры базы данных на этапе концептуального проектирования, а также избежать излишней детализации структуры *PDM*-системы на этапе ее проектирования.

### Модель предметно-ориентированной онтологии

Предметно-ориентированная онтология – это концептуализация мира в форме словаря объектов, их качественных характеристик, отличительных особенностей для данной предметной области [9, 10]. Понятия, представленные в словаре, – это принятая в данной предметной области терминология. Формально предметно-ориентированная онтология ( $O$ ) определяется следующим образом:

$$O = \langle C, I, E, L, F_{ext}, F_{md}, F_s, F_{exc} \rangle,$$

где  $C$  – набор категорий (классов объектов, понятий) единого информационного пространства;  $I$  – набор информационных ресурсов, которые описываются различными категориями;  $E$  – унифицированный обменный формат единой модели данных, который позволяет формально описать все виды разнородных информационных ресурсов;  $L$  – набор внешних форматов представления данных систем;  $F_{ext}(L_i) \rightarrow E$  – функция преобразования данных из внешних форматов в унифицированный формат единой модели данных;  $F_{md}(E) \rightarrow I$  – функция преобразования данных из обменного формата в набор информационных ресурсов;  $F_s(I) \rightarrow I$  – функция поиска требуемых информационных ресурсов ( $I$ ), удовлетворяющих заданным поисковым критериям ( $I_j$ );  $F_{exc}(I_i) \rightarrow I_j$  – функция информационного обмена между узлами ЕИП.

Понятие категории формально можно описать так:

$$C = \langle s, A, R, T \rangle,$$

где  $s = \langle s_1, s_2 \rangle$ , при этом  $s_1$  – название категории,  $s_2$  – тип категории;  $A$  – набор атрибутов категории, при этом  $A = \{a_i \mid a_i \text{ – описание } i\text{-го атрибута категории, } i = 1 \div n, n \text{ – количество атрибутов в категории}\}$ ;  $R$  – набор вложенных рубрик категории, при этом  $R = \{r_i \mid r_i \text{ – описание } i\text{-й рубрики, } i = 1 \div m, m \text{ – количество рубрик в категории}\}$ ;  $T$  – набор шаблонов визуализации категории, при этом  $T = \{t_i \mid t_i \text{ – описание } i\text{-го шаблона, } i = 1 \div k, k \text{ – количество шаблонов в категории}\}$ .

Понятие информационного ресурса определяется как

$$I = \langle p, C, V, D, S, F \rangle,$$

где  $p = \langle p_1, p_2, p_3, p_4 \rangle$ , при этом  $p_1$  – название информационного ресурса,  $p_2$  – его состояние,  $p_3$  – дата и время его создания,  $p_4$  – информация об авторе и владельце;  $C$  – категория информационного ресурса;  $V$  – набор значений атрибутов, при этом  $V = \{v_i \mid v_i \text{ – описание } i\text{-го значения атрибута, } i = 1 \div n, n \text{ – количество значений атрибутов в информационном ресурсе}\}$ ;  $D$  – набор описаний прав доступа к информационному ресурсу, при этом  $D = \{d_i \mid d_i \text{ – описание } i\text{-го права доступа некоторого пользователя, } i = 1 \div m, m \text{ – количество определенных прав доступа}\}$ ;  $d_i = \langle d_1, d_2, d_3, d_4 \rangle$ , при этом  $d_1$  – возможность чтения,  $d_2$  – возможность изменения,  $d_3$  – возможность удаления,  $d_4$  – возможность передачи информационного ресурса другим пользователям и на другие узлы системы по каналам информационного обмена;  $S$  – набор настроек синхронизации, где каждая настройка определяет достаточные данные для осуществления информационного обмена, т. е. для выполнения функции  $F_{exc}(I_i) \rightarrow I_j$ , при этом  $S = \{s_i \mid s_i \text{ – описание настройки синхронизации с } i\text{-м узлом ЕИП, } i = 1 \div k, k \text{ – количество узлов, с которыми синхронизируется информационный ресурс}\}$ ;  $F$  – набор прикрепленных файлов, при этом  $F = \{f_i \mid f_i \text{ – описание } i\text{-го файла, } i = 1 \div l, l \text{ – количество прикрепленных к информационному ресурсу файлов}\}$ .

В разрабатываемой системе все циркулирующие в процессе проектирования СБИС СнК данные можно разделить на пять основных категорий:

- цифровая модель (например, модель в формате *MATLAB/Simulink*);
- исходный код (например, описание устройства на языке *VHDL*);
- отчет (например, отчет о физической верификации топологии кристалла);
- текстовый документ (например, текст спецификации);
- проектный электронный документ (все остальные документы, не подпадающие под вышеперечисленные типы, например, файл описания топологии).

Каждой категории соответствует свой тип файлов – файл цифровой модели, файл исходного кода, файл отчета, файл текстового документа, файл проектного электронного документа. В соответствии с ГОСТ 2.051-2006 электронный конструкторский документ организован в виде реквизитной части и одной или нескольких содержательных частей. Реквизитная часть включает в себя описания конструкторского документа, а содержательная часть – соответствующие ему файлы. Содержимое реквизитной части регламентируется ГОСТ 2.051-2006 и ГОСТ 2.104-2006.

Информационный ресурс в терминах предметно-ориентированной онтологии состоит из реквизитной части (параметры  $p, C, V, D, S$ ) и содержательной части (параметр  $F$ ).

Атрибуты различных типов файлов содержательной части (табл. 1–3) в целом аналогичны, но имеются некоторые специфические различия, в частности наличие атрибутов описания программного обеспечения-симулятора. Любой атрибут может заполняться либо пользователем в



процессе работы с системой вручную, либо автоматически, либо автоматически при загрузке соответствующего файла в хранилище из интерфейса программного обеспечения-редактора, либо вручную путем выбора из списка доступных значений атрибута.

Таблица 1. Атрибуты файла типа «Цифровая модель»

| Атрибут                     | Способ заполнения                   |
|-----------------------------|-------------------------------------|
| Имя файла                   | Вручную                             |
| Расширение файла            | Автоматически                       |
| Размер файла                | Автоматически                       |
| Тип файла в терминах ПЦ     | Вручную, из предоставленного списка |
| ПО-редактор – наименование  | Автоматически, при выгрузке из ПО   |
| ПО-редактор – версия        | Автоматически, при выгрузке из ПО   |
| ПО-симулятор – наименование | Автоматически, при выгрузке из ПО   |
| ПО-симулятор – версия       | Автоматически, при выгрузке из ПО   |
| Язык описания               | Автоматически, при выгрузке из ПО   |

Таблица 2. Атрибуты файла типа «Отчет»

| Атрибут                    | Способ заполнения                   |
|----------------------------|-------------------------------------|
| Имя файла                  | Вручную                             |
| Расширение файла           | Автоматически                       |
| Размер файла               | Автоматически                       |
| Тип файла в терминах ПЦ    | Вручную, из предоставленного списка |
| ПО-редактор – наименование | Автоматически, при выгрузке из ПО   |
| ПО-редактор – версия       | Автоматически, при выгрузке из ПО   |

Таблица 3. Атрибуты файла типа «Проектный электронный документ»

| Атрибут                     | Способ заполнения                   |
|-----------------------------|-------------------------------------|
| Имя файла                   | Вручную                             |
| Расширение файла            | Автоматически                       |
| Размер файла                | Автоматически                       |
| Тип файла в терминах ПЦ     | Вручную, из предоставленного списка |
| ПО-редактор – наименование  | Автоматически, при выгрузке из ПО   |
| ПО-редактор – версия        | Автоматически, при выгрузке из ПО   |
| ПО-симулятор – наименование | Автоматически, при выгрузке из ПО   |
| ПО-симулятор – версия       | Автоматически, при выгрузке из ПО   |
| Язык описания               | Автоматически, при выгрузке из ПО   |
| Аппаратная платформа        | Вручную, из предоставленного списка |
| Блок верхнего уровня        | Вручную                             |

Реквизитная часть представлена индивидуальной карточкой документа. Набор атрибутов карточки документа зависит от того, куда передается или откуда получен данный документ. Список доступных атрибутов представлен в табл. 4.

Таблица 4. Атрибуты карточки документа

| Атрибут  | Способ заполнения                   |
|--|-------------------------------------|
| Наименование документа   | Вручную                             |
| Наименование изделия   | Вручную, из предоставленного списка |
| Обозначение документа  | Вручную, из предоставленного списка |
| Код документа  | Автоматически                       |
| Наименование организации   | Автоматически                       |
| Код документа, взамен или на основании которого<br>выпущен данный документ | Вручную                             |
| Стадия ЖЦ документа  | Вручную, из предоставленного списка |
| Версия документа   | Автоматически                       |
| Текстовое описание документа   | Вручную                             |
| Уровень конфиденциальности документа                                       | Вручную, из предоставленного списка |
| Дата внесения изменений  | Автоматически                       |
| Краткое описание изменений   | Вручную                             |
| Порядковый номер изменения   | Автоматически                       |
| Электронная подпись лица, внесшего изменение                               | Вручную                             |
| ФИО лица, подписавшего документ  | Автоматически                       |
| Характер работы лица, подписавшего документ                                | Автоматически                       |
| Электронная подпись лица, подписавшего<br>документ                         | Вручную                             |
| Дата подписания документа  | Автоматически                       |
| ФИО лица, осуществившего последнюю загрузку<br>документа                   | Автоматически                       |
| Характер работы лица, осуществившего<br>последнюю загрузку документа       | Автоматически                       |
| Дата последней загрузки документа из хранилища                             | Автоматически                       |
| Дополнительные атрибуты  |                                     |
| Передача документа с одного этапа проектного цикла на другой               |                                     |
| Идентификатор пользователя-получателя<br>документа                         | Вручную, из предоставленного списка |
| Идентификатор предыдущего этапа ПЦ   | Вручную, из предоставленного списка |
| Идентификатор последующего этапа ПЦ  | Вручную, из предоставленного списка |
| Получение документа от внешнего поставщика                                 |                                     |
| Идентификатор внешнего поставщика документа                                | Вручную, из предоставленного списка |
| Передача документа внешнему получателю                                     |                                     |
| Идентификатор внешнего получателя документа                                | Вручную, из предоставленного списка |
| Передача документа в рамках текущего этапа проектного цикла                |                                     |
| Идентификатор пользователя-получателя<br>документа                         | Вручную, из предоставленного списка |
| Идентификатор этапа ПЦ   | Вручную, из предоставленного списка |

Как и при работе с файлами, любой атрибут может либо заполняться пользователем вручную, либо заполняться автоматически, либо заполняться вручную путем выбора из списка доступных значений атрибута.

### **Описание алгоритмов работы функций по управлению основными сущностями онтологии**

В состав модели онтологии единого информационного пространства проектирования СБИС СнК входят различные сущности и элементы словаря онтологии, а также набор функций по управлению этими сущностями:

- 1) функция преобразования данных из внешних форматов в унифицированный обменный формат единой модели данных ( $F_{ext}(L_i) \rightarrow E$ );
- 2) функция преобразования данных из обменного формата в набор информационных ресурсов ( $F_{md}(E) \rightarrow I$ );
- 3) функция поиска информационных ресурсов в соответствии с заданными критериями ( $F_s(I_i) \rightarrow I_j$ );
- 4) функция информационного взаимодействия узлов ЕИП ( $F_{exc}(I_i) \rightarrow I_j$ ).

На программно-аппаратном уровне ЕИП каждая из этих функций должна быть представлена набором программных модулей, которые реализуют их алгоритмы.

Функции преобразования данных  $F_{ext}(L_i) \rightarrow E$  и  $F_{md}(E) \rightarrow I$  осуществляют преобразования конструкторских документов в соответствии с положениями ГОСТ 2.051-2006 и ГОСТ 2.104-2006 (создание и редактирование реквизитной и содержательных частей информационного ресурса).

Функция ( $F_{exc}(I_i) \rightarrow I_j$ ) получает на вход набор информационных ресурсов, которые требуется передать получателю. Функция осуществляет анализ адресов получателей и посредством программной платформы информационного взаимодействия производит доставку наборов информационных ресурсов на целевые узлы. В основе работы данной функции лежат следующие основные требования:

- 1) наличие единых правил присвоения адресов субъектам и объектам ЕИП;
- 2) согласование содержания информационного и информационно-лингвистического обеспечения отдельных объектов ЕИП;
- 3) поддержание объектов ЕИП в актуальном состоянии;
- 4) гарантированное доведение информационных ресурсов до потребителей, в том числе при «нестабильных» каналах связи.

На рис. 2 представлена диаграмма состояний подсистемы информационного взаимодействия узлов ЕИП ( $F_{exc}(I_i) \rightarrow I_j$ ).

Укрупненно поведение подсистемы информационного взаимодействия определяется шестью состояниями:

1. Получение от вызывающей функции (например, функции поиска  $F_s(I_i) \rightarrow I$ ) набора передаваемых информационных ресурсов.
2. Анализ адресов получателей, их сопоставление с зарегистрированными в системе адресатами. В данном состоянии выбирается очередной адресат, если он доступен, и осуществляется переход в следующее состояние. В противном случае проверка доступности

выполняется для следующего адресата в списке до тех пор, пока не будут обработаны все адресаты.

3. Формирование контрольной суммы передаваемого пакета.

4. Доставка пакета по каналам связи с использованием алгоритма циклического избыточного кода.

5. Ожидание ответа от адресата о получении передаваемого пакета. Предполагается, что адресат должен отправить ответ только в случае успешного получения пакета и совпадения эталонной и вычисленной контрольных сумм. Если ответ от адресата не получен и превышен порог ожидания ответа, то выполняется переход в состояние 3. Если ответ получен, то выполняется передача пакета следующему адресату. Состояния 3–5 выполняются для всех указанных адресатов, при этом их выполнение осуществляется в асинхронном режиме.

6. Завершение работы и передача управления в вызвавшую функцию.

Переходы между состояниями осуществляются автоматически, о окончании внутренней деятельности состояния или по истечении отведенного времени.

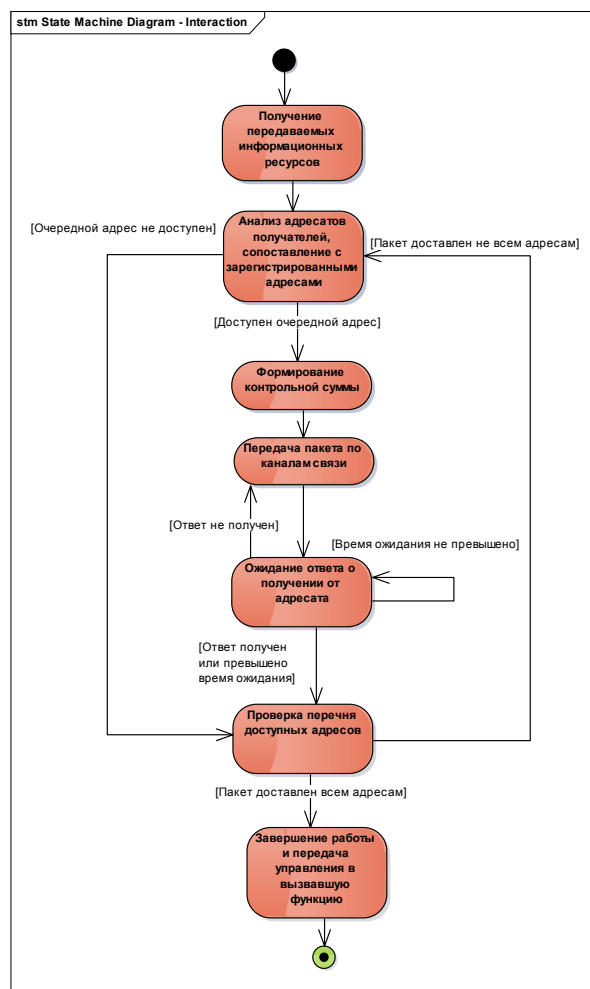


Рис. 2. Диаграмма состояний подсистемы информационного взаимодействия узлов ЕИП

Функция поиска ( $F_s(I_i) \rightarrow I$ ) реализована таким образом, чтобы обеспечить распределенный поиск требуемых информационных ресурсов на всех узлах ЕИП с учетом прав доступа пользователей. На рис. 3 представлена диаграмма состояний подсистемы, реализующей данную функцию. Поведение подсистемы определяется одиннадцатью состояниями:

1. Формирование критериев поиска в виде эталонного информационного ресурса  $I_i$ . Поиск критерии задаются пользователем, выполняющим поиск. Функция агрегирует эти критерии в виде набора значений атрибутов, входящих в категорию формируемого эталонного информационного объекта.

2. Определение прав доступа пользователя к механизму поиска. По умолчанию полагается, что пользователь не имеет прав на искомый информационный ресурс, если не указано обратное. При наличии прав осуществляется поиск, при их отсутствии пользователь получает соответствующее информационное сообщение.

3. Поиск требуемой информации в базе данных текущего узла ЕИП, при этом происходит генерация *SQL*-запросов к таблицам базы данных, в которых хранятся информационные ре-

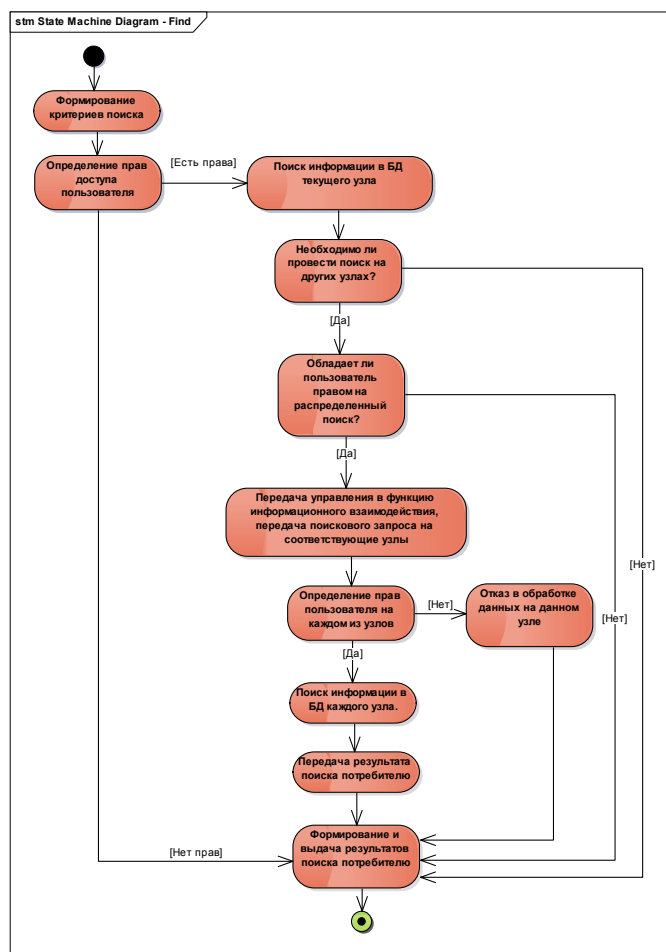


Рис. 3. Диаграмма состояний подсистемы поиска информационных ресурсов

сурсы. Результатом выполнения *SQL*-запросов является промежуточный набор информационных ресурсов, которые удовлетворяют запросу пользователя.

4. Определение необходимости поиска на других узлах ЕИП. Если принимается решение об отсутствии необходимости поиска на удаленных узлах, то осуществляется переход в состояние 11, где формируется окончательный ответ пользователю. В противном случае управление передается в состояние 5.

5. Определение прав доступа пользователя к механизму поиска на других узлах. По умолчанию полагается, что пользователь не имеет прав на поиск на других узлах, если не указано обратное. При наличии прав осуществляется поиск, при их отсутствии пользователь получает соответствующее информационное сообщение.

6. Передача сформированного на первом шаге эталонного информационного ресурса  $I_i$  в функцию информационного взаимодействия ( $F_{exc}(I_i) \rightarrow I_j$ ), которая осуществляет его передачу на все требуемые узлы ЕИП.

7. Определение полномочий пользователя, выдавшего поисковый запрос, на каждом из узлов ЕИП, на которые был доставлен эталонный информационный ресурс  $I_i$ . По умолчанию предполагается, что пользователь не имеет прав на потенциально искомый информационный ресурс, если не указано обратное. Если пользователю не выданы полномочия на поиск на данном узле, то происходит переход в состояние 8. В противном случае происходит переход в состояние 9.

8. Формирование отказа от выполнения поиска на данном удаленном узле, формирование пустого набора информационных ресурсов в качестве ответа.

9. Генерация *SQL*-запросов к таблицам базы данных, в которых осуществляется хранение информационных ресурсов. Результатом выполнения *SQL*-запросов является промежуточный набор информационных ресурсов, которые удовлетворяют запросу пользователя.

10. Передача сформированного промежуточного набора информационных ресурсов в функцию информационного взаимодействия ( $F_{exc}(I_i) \rightarrow I_j$ ), которая осуществляет передачу результатов поиска потребителю.

11. Ожидание завершения работы функции информационного взаимодействия, которая должна доставить ответы от всех удаленных узлов, где осуществлялся поиск. После этого с учетом полученных ответов, а также промежуточного набора информационных ресурсов, сформированных на втором шаге, потребителю выдаются результаты поиска.

Как и в предыдущем случае, переходы между состояниями осуществляются автоматически, при окончании внутренней деятельности состояния или по истечении отведенного времени.

В качестве примера структуры *PDM*-системы, поддерживающей проектный цикл СБИС СнК, рассмотрим подсистему информационной поддержки этапа логического проектирования. Данная подсистема реализована с учетом модели предметной области, полученной с использованием аппарата предметно-ориентированных онтологий, рассмотренных ранее. Логическая структура и поведение предлагаемой подсистемы разработаны с учетом единой понятийной среды, реализуют единые правила адресации субъектов и объектов ЕИП. В подсистеме предусмотрены механизмы согласования и поддержания актуальности информационного обеспечения.

Структурная схема этапа логического проектирования, получаемых, генерируемых и передаваемых данных-документов об изделии представляется в виде линейно организованной структуры. У каждого типа документа такой структуры создается статус происхождения: «извне», «создается», «создается/передается» (табл. 4). Документы могут быть представлены в различных форматах файлов, в частности .v (*RTL*-описание на языке *Verilog HDL*), .vhd (*RTL*-описание на языке *VHDL*), .csv (текстовый формат представления табличных данных) и др.

Для каждого документа, формируемого на этапе проектирования СБИС СнК, можно выделить последовательность стадий его собственного жизненного цикла, что отражается в свойствах документа в *PDM*-системе как «Текущий статус» (табл. 5).

На начальном этапе своего ЖЦ документ создается в *PDM*-системе в состоянии «Новый». Как только по данному документу запущены процессы проектирования, он переходит в состояние «В разработке». Документ считается готовым к передаче в случае завершения всех связанных с ним процессов проектирования и готовности документа либо к передаче на следующий этап жизненного цикла проектирования, либо непосредственно на производство. Если документ не находится в стадии разработки или изготовления и не является «Новым», то состояние его ЖЦ меняется на «Неактивный».

Параметр «Текущий статус» относится, прежде всего, к документам, генерируемым на текущем этапе проектирования, не выходящим за пределы текущего этапа и, таким образом, не требующим дополнительного согласования и подготовки конструкторской документации в бумажном виде.

Таблица 4. Документы этапа логического проектирования

| Порядковый номер | Тип документа                                   | Статус документа     |
|------------------|---|----------------------|
| 1                | Детальная спецификация СнК                      | Извне                |
| 2                | Спецификация интерфейсов                        | Извне                |
| 3                | Системные ограничения                           | Извне                |
| 4                | <i>RTL</i> -описание СнК                        | Извне                |
| 5                | Файлы тестового окружения СнК                   | Извне                |
| 6                | Стандартная библиотека производства             | Извне                |
| 7                | Результаты физической оптимизации               | Создается            |
| 8                | Результаты статического временного анализа      | Создается            |
| 9                | Результаты формальной верификации               | Создается            |
| 10               | Результаты моделирования списка соединений      | Создается            |
| 11               | Список соединений ( <i>Netlist</i> )            | Создается/передается |
| 12               | Предварительные параметры кристалла             | Создается/передается |
| 13               | Предварительные характеристики СнК              | Создается/передается |
| 14               | Описание временных ограничений                  | Создается/передается |
| 15               | Ограничения на расположение компонентов проекта | Создается/передается |

Таблица 5. Стадии жизненного цикла элементов в терминах *PDM*

| Стадия ЖЦ документа («Текущий статус») | Описание  |
|--|---|
| Новый                                  | На начальном этапе проектирования (при создании) элемент находится в состоянии «Новый»    |
| Разработка                             | По элементу запущен процесс проектирования  |
| Готов к передаче                       | Элемент готов к запуску в производство либо к переходу на следующий этап жизненного цикла |
| Неактивный                             | Элемент не находится в стадии разработки или изготовления (производства)                  |

Таблица 6. Стадии жизненного цикла объектов проектирования

| Тип объекта                         | Порядковый номер стадии | Стадия жизненного цикла |
|-------------------------------------|-------------------------|-------------------------|
| Информационная модель изделия (ИМИ) | 1                       | Новый                   |
|                                     | 2                       | Опубликован             |
|                                     | 3                       | На редактировании       |
|                                     | 4                       | Новая версия            |
|                                     | 5                       | Выпущен                 |
|                                     | 6                       | Согласован              |
|                                     | 7                       | Оригинал                |
| Конструкторская документация (КД)   | 1                       | Новый                   |
|                                     | 2                       | Опубликован             |
|                                     | 3                       | Новая версия            |
|                                     | 4                       | Выпущен                 |
|                                     | 5                       | Согласован              |
|                                     | 6                       | Подлинник               |

Документы, передаваемые на последующие этапы жизненного цикла и непосредственно на производство, можно подразделить на два больших класса: информационная модель изделия (*RTL*-описание, список соединений) и конструкторская документация (отчеты и спецификации разного рода), стадии жизненного цикла которых представлены в табл. 6.

Собственно *PDM*-система имеет древовидную иерархическую структуру, на верхнем уровне которой находится собственно разрабатываемая СБИС СнК, на нижних уровнях – документы, полученные в ходе процесса проектирования отдельных СФ-блоков, и документы, полученные в ходе процесса проектирования собственно СБИС СнК (описания, отчеты, файлы исходного кода). В зависимости от текущего этапа проектирования и должности специалиста, запрашивающего доступ к базе данных, возможна фильтрация отображаемых и доступных для чтения и редактирования документов.

Реализация данной структуры может быть выполнена как на базе машиностроительных *PDM*-систем после соответствующей модификации, так и на базе систем контроля версий *Rational ClearCase*, *Perforce* и *PVCS* [15].



## Заключение

Использование предметно-ориентированных онтологий при проектировании систем управления данными позволяет избежать информационной противоречивости сложных многосвязных структур и организовать такую систему управления и хранения, которую возможно реализовать с учетом разграничения прав пользователей. Такое разграничение, заложенное уже на концептуальном уровне проектирования, позволит избежать информационной избыточности, возникающей при прямом использовании унифицированных *PDM*-систем в микроэлектронном производстве. Такая избыточность как затрудняет визуальное считывание проектной информации сотрудниками, так и резко нагружает сетевую инфраструктуру предприятия при ведении процессов проектирования.

В статье проанализированы этапы проектирования и передачи в производство изделий микроэлектроники на основе заказных или полузаказных сверхбольших интегральных схем класса «Система на кристалле». Рассмотрена структура системы управления данными об изделии, являющейся ключевым элементом единого информационного пространства, отвечающего требованиям микроэлектронного производства. Анализ и проектирование предлагаемой структуры *PDM*-системы выполнены с использованием положений предметно-ориентированной онтологии.

На основе результатов проведенного анализа сформулированы требования к составу и содержанию документов, входящих в информационное обеспечение автоматизированной системы управления процессами жизненного цикла микроэлектронного производства, имеющей иерархическую структуру. Результаты исследования могут быть использованы при разработке архитектуры системы управления данными электронных изделий класса «Система на кристалле» для интеграции всех этапов жизненного цикла и организации единого информационного пространства проектной или проектно-производственной организации.

## Список литературы

- [1] Немудров В., Мартин Г. *Системы-на-кристалле. Проектирование и развитие*. М.: Техносфера, 2004. 2016 с. [Nemudrov V., Martin G. *System-on-Chip. Design and development*. Moscow, Tekhnosfera, 2004, 2016 p. (in Russian)]
- [2] Chin D. Nanoelectronics for an Ubiquitous-Information Society, *IEEE International Solid – State Circuits Conference Digest of Technical Papers*, 2005, 2, 22–26.
- [3] Карин А.С. Построение предметно-ориентированной онтологии в системах обработки пространственных данных. *Информационно-управляющие системы*, 2014, 4, 78–84. [Karin A.S. Developing a domain-specific ontology in spatial data processing systems, *Information and Control Systems*, 2014, 4, 78–84. (in Russian)]
- [4] Куликов Г.Г., Ризванов К.А., Христюбов В.Л. Организация единого информационного пространства для распределенного выполнения проектов в авиадвигателестроении. *Вестник УГАТУ. Управление в социальных и экономических системах*, 2012, 16 (6), 202–210. [Kulikov G.G., Rizvanov K.A., Khristolyubov V.L. Organization of integrated information environment for the distributed execution of projects in the aircraft engine, *Bulletin USATU. Management in social and economic systems*, 2012, 16 (6), 202–210. (in Russian)]

- [5] Шпилина Е., Петров А. Опыт работы ПО «Севмаш» по внедрению ИТ-технологий. *Рациональное управление предприятием*, 2011, 3, 40–42. [Shpilina Ye., Petrov A. Experience JSCo «PO «Sevmash» on the introduction of IT technology, *Rational Enterprise Management*, 2011, 3, 40–42. (in Russian)]
- [6] Шабалкин Д.Ю. Интеграция полиплатформенных автоматизированных подсистем различной функциональности в единое информационное пространство жизненного цикла изделия авиационной техники. *Известия Самарского научного центра Российской академии наук*, 2012, 14 (4), 545–549. [Shabalkin D.Yu. The integration of polyplatform systems automated of various functionality to uniform information space of the product life cycle of aircraft, *Proceedings of the Samara Scientific Center of the Russian Academy of Sciences*, 2012, 14 (4), 545–549. (in Russian)]
- [7] Борискин В.С., Гулякович Г.Н., Северцев В.Н. Организация мелкосерийного производства микросхем. *Инженерный вестник Дона*, 2012, 20 (2), 310–314. [Boriskin V.S., Gulyakovich G.N., Severtsev V.N. Organization of small batch production of chips, *Engineering journal of Don*, 2012, 20 (2), 310–314. (in Russian)]
- [8] Лапшин В.А. *Онтологии в компьютерных системах*. М.: Научный мир, 2010. 224 с. [Lapshin V.A. *Ontologies in computer systems*. Moscow, Nauchnyy mir, 2010, 224 p. (in Russian)]
- [9] Соловьев В.Д., Добров Б.В., Иванов В.В., Лукашевич Н.В. *Онтологии и тезаурусы: модели, инструменты, приложения*. М.: Бином. Лаборатория знаний, 2009. 173 с. [Solovev V.D., Dobrov B.V., Ivanov V.V., Lukashevich N.V. *Ontologies and thesauruses: models, tools, application*. Moscow: Binom. Laboratoriya znaniy, 2009, 173 p. (in Russian)]
- [10] Найханова Л.В. *Технология создания методов автоматического построения онтологий с применением генетического и автоматного программирования: монография*. Улан-Удэ: БНЦ СО РАН, 2008. 244 с. [Naykhanova L.V. *Technology for creating ontology automatic construction methods using genetic programming and of automata: a monograph*. Ulan-Ude, BNTs SO RAN, 2008, 244 p. (in Russian)]
- [11] Новичков С. Маршрут разработки СБИС типа «система на кристалле». *Электронные компоненты*, 2006, 12, 63–67. [Novichkov S. The development route of VLSI-type «System on a Chip», *Electronic components*, 2006, 12, 63–67. (in Russian)]
- [12] Мосин С.Г., Кухарук В.С., Фёдоров С.В. Маршрут проектирования цифровых ЗИС в САПР Mentor Graphics. *Проектирование и технология электронных средств*, 2006, 1, 9–12. [Mosin S.G., Kukharuk V.S., Fedorov S.V. The design route of digital custom ICs on CAD tools of Mentor Graphics, *Design and technology of electronic means*, 2006, 1, 9–12. (in Russian)]
- [13] Судов Е.В., Левин А.И. *Концепция развития CALS-технологий в промышленности России*. М.: НИЦ CALS-технологий «Прикладная логистика», 2002. 129 с. [Sudov Ye.V., Levin A.I. *The concept of CALS-technologies in Russian industry*. Moscow, NITs CALS-tekhnologiy «Prikladnaya logistika», 2002, 129 p. (in Russian)]
- [14] Okino N. Object and Operation dualism for CAD/CAM architecture, *Annals of the CIRP*, 1983, 34 (1), 179–182.
- [15] Simpson P. *FPGA Design. Best Practices for team-based Design*. New York, Springer, 2010, 151 p.